SYSTEM SECURITY

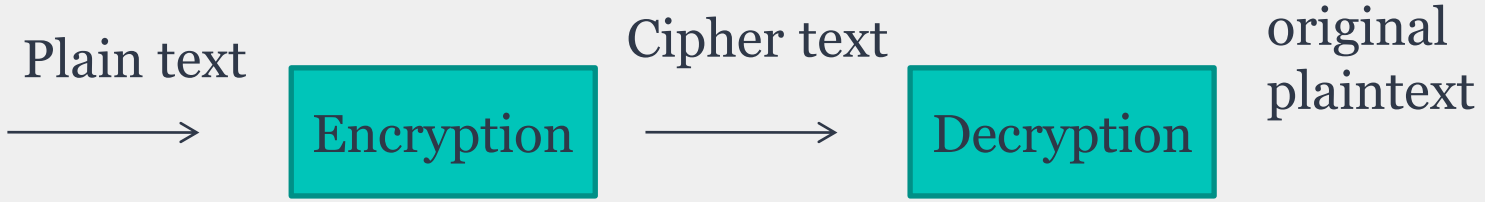
UNIT 3: CRYPTOGRAPHY

By Prof. Krishna Samdani

CRYPTOGRAPHY BASICS

- A message is ***plaintext*** (sometimes called clear text).
- The process of disguising a message in such a way as to hide its substance is ***encryption***.
- An encrypted message is ***cipher text***.
- The process of turning cipher text back into plaintext is ***decryption***.
- The art and science of keeping messages secure is ***cryptography***, and it is practiced by cryptographers.
- ***Cryptanalysts*** are practitioners of cryptanalysis, the art and science of breaking cipher text; that is, seeing through the disguise.
- The branch of mathematics encompassing both cryptography and cryptanalysis is ***cryptology*** and its practitioners are cryptologists

CRYPTOGRAPHY BASICS



- The encryption function E, operates on M to produce C. Or, in mathematical notation:

$$E(M) = C$$

- In the reverse process, the decryption function D operates on C to produce M:

$$D(C) = M$$

$$D(E(M)) = M$$

CRYPTOGRAPHY BASICS

- A ***cryptographic algorithm***, also called a ***cipher***, is the mathematical function used for encryption and decryption.
- Both the encryption and decryption operations use a key. The range of possible values of the key is called the ***key space*.**
- $E_K(M) = C$
- $D_K(C) = M$
- $D_K(E_K(M)) = M$
- A ***cryptosystem*** is an algorithm, plus all possible plaintexts, ciphertexts, and keys**.**

SUBSTITUTION CIPHERS

- A ***substitution cipher*** is one in which each character in the plaintext is substituted for another character in the cipher text. The receiver inverts the substitution on the cipher text to recover the plaintext.

CAESAR CIPHER

- Each character of a message is replaced by a character *three position down in the alphabet*.
- If we represent each letter of the alphabet by an integer that corresponds to its position in the alphabet, the formula for replacing each character ***p*** of the plaintext with a character ***c*** of the cipher text can be expressed as
- ***c = (p + 3) mod 26***

Example:

Plain text: hello w o r l d

p= h=8 so

c= (p+3)%26= (8+3)%26 = 11= k

Cipher text: khoor zruog

Practice Question

- **Cipher Text** - lwv mxvw wkh ehjlqqlqj
- **Key** – 3
- **Plain Text** -

MODIFIED VERSION OF CAESAR CIPHER

- In this, the *original plain text alphabets* may not necessarily be replaced three places down the line, but instead *can be replaced any places down the line.*

- Once the replacement scheme is decided, it would be constant and will be used for all the other alphabets in that message.

- Example : Alphabet A in the plain text would not necessarily be replaced by D. It can be replaced by any valid alphabet i.e by E or F or by G and so on.

Practice Question

- **Cipher Text** - oek xqlu q juij ed iqjkhtqo
- **Key** – 16
- **Plain Text** -

MONO-ALPHABETIC CIPHER

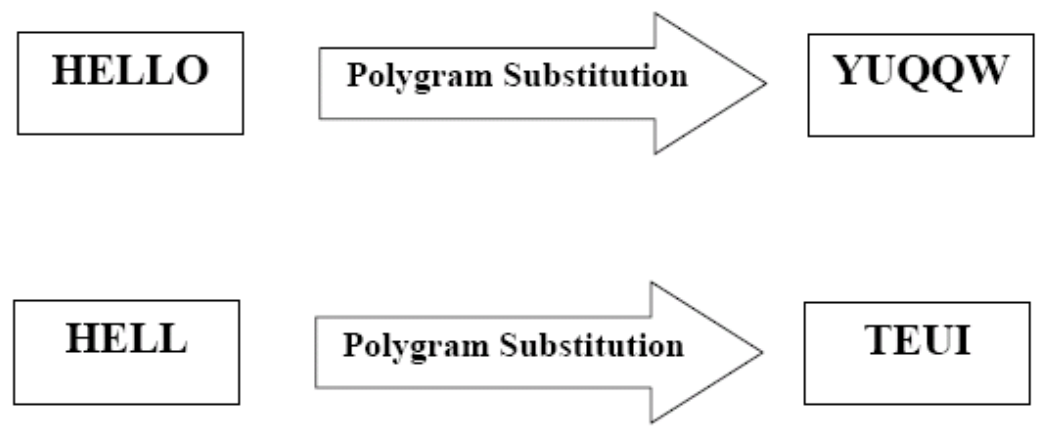
- The mono-alphabetic substitution takes a letter of an alphabet and substitutes it with another letter, this way a cipher text is generated. The way of converting is fixed. A character of the plaintext will be replaced by the same cipher text character, during the entire cipher text. There is no additional key. The only way of security is to keep the substitution-table secret.
- It uses random substitution.
- Example: For any given plain text message, each A in the message can be replaced by any other alphabet (B through Z). Likewise, each B can be replaced by any other random alphabet & so on.
- There is no relation between the replacement of A & replacement of B.

HOMOPHONIC SUBSTITUTION CIPHER

- *In this, one plain text alphabet can map to more than one cipher text alphabet.*
- Example: A can be replaced by D, H, P, R.

POLYGRAM SUBSTITUTION CIPHER

- **Polygram substitution cipher** is one in which *blocks of characters are encrypted* in groups.



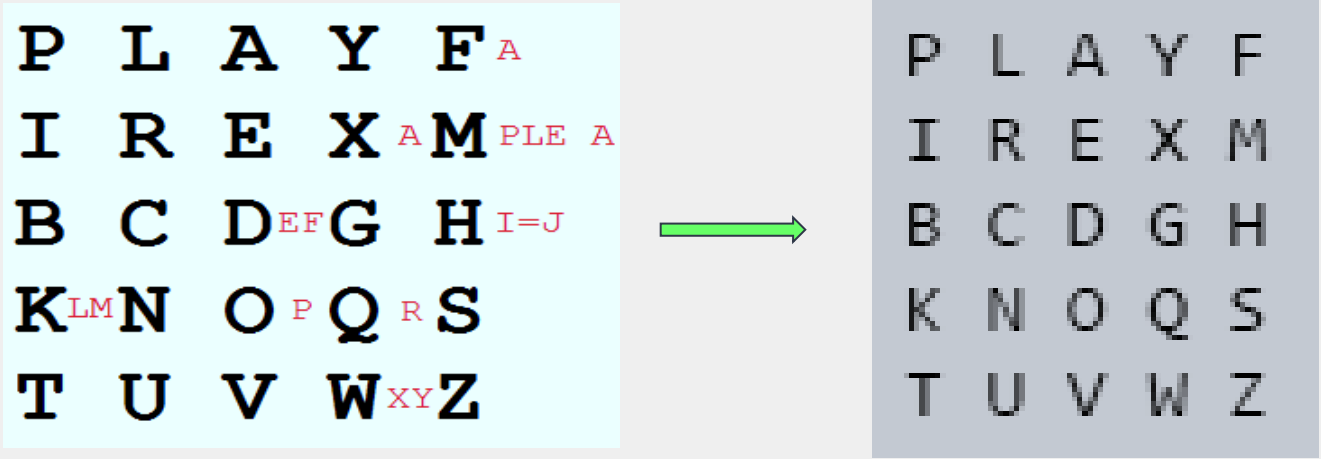
PLAYFAIR CYPHER

The Playfair cipher uses a 5 by 5 table containing a key word or phrase. Memorization of the keyword and 4 simple rules was all that was required to create the 5 by 5 table and use the cipher.

1. If both letters are the same (or only one letter is left), add an "X" after the first letter. Encrypt the new pair and continue. Some variants of Playfair use "Q" instead of "X", but any letter, itself uncommon as a repeated pair, will do.
2. If the letters appear on the same row of your table, replace them with the letters to their immediate right respectively (wrapping around to the left side of the row if a letter in the original pair was on the right side of the row).
3. If the letters appear on the same column of your table, replace them with the letters immediately below respectively (wrapping around to the top side of the column if a letter in the original pair was on the bottom side of the column).
4. If the letters are not on the same row or column, replace them with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. The order is important – the first letter of the encrypted pair is the one that lies on the same **row** as the first letter of the plaintext pair.

PLAYFAIR CYPHER

- Using Key: "***playfair example***" (assuming that I and J are interchangeable)

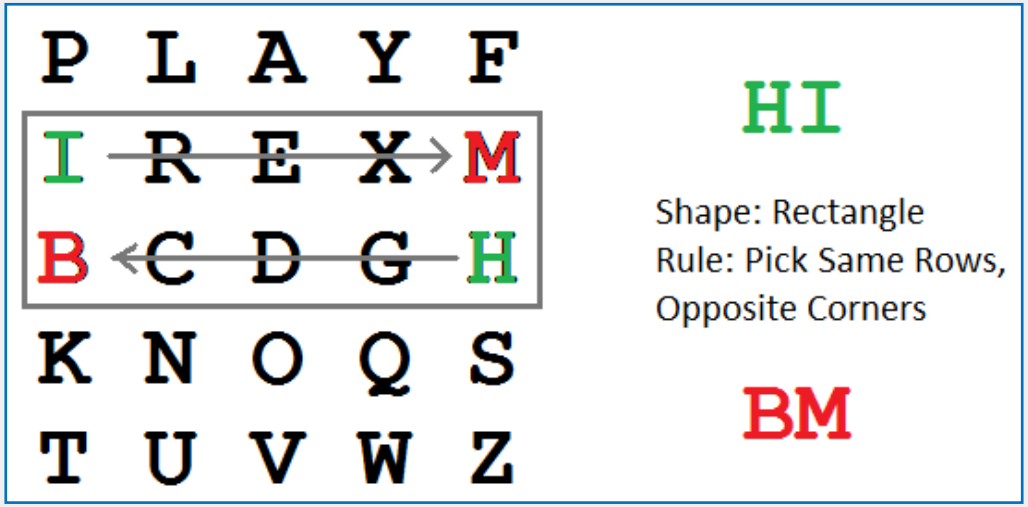


PLAYFAIR CYPHER

Encrypting the message "***Hide the gold in the tree stump***"

(note the null "X" used to separate the repeated "E"s) :

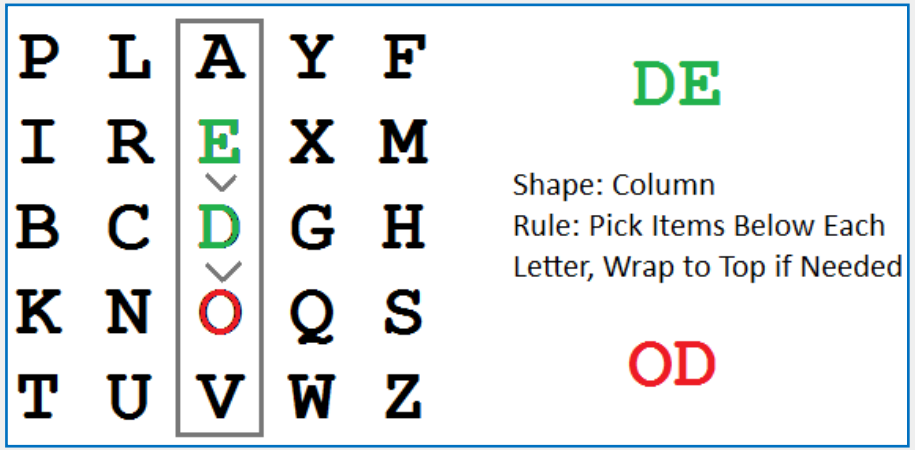
STEP 1:



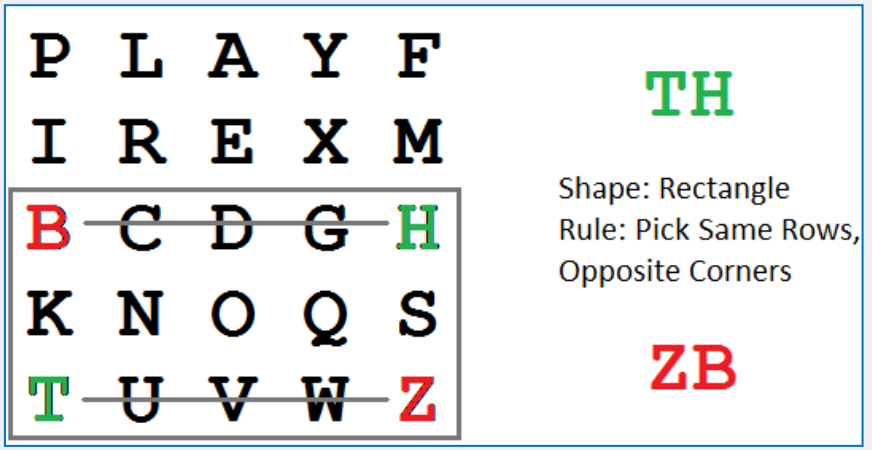
PLAYFAIR CYPHER

HI DE TH EG OL DI NT HE TR EX ES TU MP
^

STEP 2:



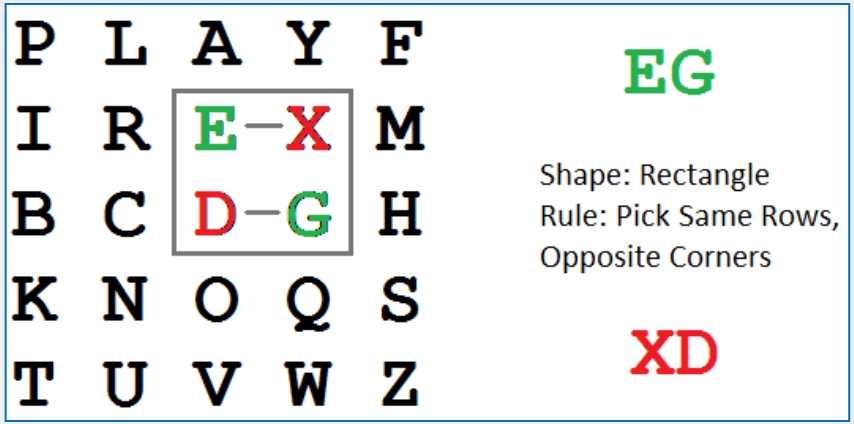
STEP 3:



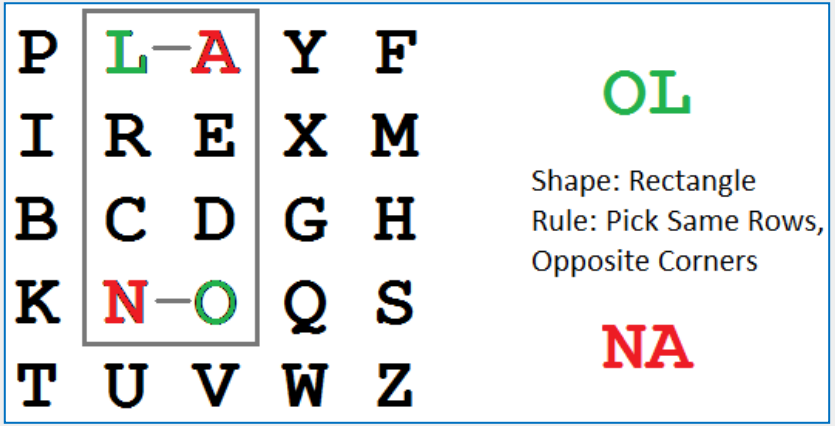
PLAYFAIR CYPHER

HI DE TH EG OL DI NT HE TR EX ES TU MP
^

STEP 4:



STEP 5:



PLAYFAIR CYPHER

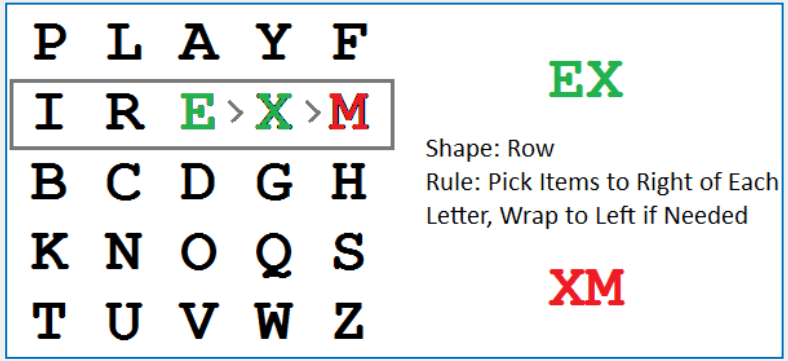
STEP 6 to 9:

6. The pair DI forms a rectangle, replace it with BE
7. The pair NT forms a rectangle, replace it with KU
8. The pair HE forms a rectangle, replace it with DM
9. The pair TR forms a rectangle, replace it with UI

STEP 11 to 13:

11. The pair ES forms a rectangle, replace it with MO
12. The pair TU is in a row, replace it with UV
13. The pair MP forms a rectangle, replace it with IF

STEP 10:



PLAYFAIR CYPHER

- **Plain Text:** HI DE TH EG OL DI NT HE TR EX ES TU MP
  ^
- Key: "***playfair example***"
- **Cypher Text:** BM OD ZB XD NA BE KU DM UI XM MO UV IF
- **Decryption:**

  To decrypt, use the INVERSE (opposite) of the last 3 rules, and the 1st as: dropping any extra "X"s, that do not make sense in the final message when finished.

Practice Question

- **Cipher Text** - VGFSLYPYGQHKNFVYXNFMYDQSFYVY
- **Key** – puzzle
- **Plain Text** -

SUBSTITUTION CIPHERS

- ***Polyalphabetic substitution cipher*** is made up of multiple simple substitution ciphers. For example, there might be five different simple substitution ciphers used; the particular one used changes with the position of each character of the plaintext.
  Example: ***Vigenere cipher***

- ***Vigenere cipher*** starts with a 26 x 26 matrix of alphabets in sequence. First row starts with 'A', second row starts with 'B', etc.

- this cipher also requires a keyword that the sender and receiver know ahead of time

- Each character of the message is combined with the characters of the keyword to find the cipher text character

VIGENERE CIPHER

- Plaint Text –

attack at dawn (column)

- Key –

lemon (row)

- Keystream –

lemonlemonle

- Cipher Text –

LXFOPVEFRNHR

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	D
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	C
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	B
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Practice Question

- **Plaint Text** –
- **Key** – computer
- **Keystream** –
- **Cipher Text** – xugvx csl iyr

TRANSPOSITION CIPHERS

- A ***transposition cipher*** is obtained by performing some sort of permutation on the plain text symbols.

RAIL FENCE TECHNIQUE

Example: We encipher NOTHING IS AS IT SEEMS by first writing it on two lines in a zig-zag pattern (or *rail fence*). The ciphertext is produced by transcribing the first row followed by the second row.

```
N   T   I   G   S   S   T   E   M
  O   H   N   I   A   I   S   E   S
```

Ciphertext: NTIGS STEMO HNIAI SES.

To decrypt, we write half the letters on one line, half on the second. (Note that if there are an odd number of letters, we include the "middle" letter on the top line.)

Example: Decipher MKHSE LWYAE ATSOL.

Solution: Since there are 15 letters, we write 8 on the top line and 7 on the bottom line so that

```
M   K   H   S   E   L   W   Y
  A   E   A   T   S   O   L
```

Plaintext: MAKE HASTE SLOWLY.

SIMPLE COLUMNAR TRANSPOSITION TECHNIQUE

```
Column No.  6 3 2 4 1 5
Plaintext:  W E A R E D
            I S C O V E
            R E D F L E
            E A T O N C
            E Q K J E U
```

Key: ZEBRAS

Ciphertext:
EVLNE ACDTK ESEAQ ROFOJ
DEECU WIREE

1. Write the plain text message row-by-row in a rectangle of a pre-defined size.

2. Read the message column-by-column. However, it need not be in the order of columns 1, 2, 3 etc. It can be any random order such as 2, 3, 1, etc.

3. The message thus obtained is the cipher text message.

SIMPLE COLUMNAR TRANSPOSITION TECHNIQUE WITH MULTIPLE ROUNDS

1. Write the plain text message row-by-row in a rectangle of a pre-defined size.

2. Read the message column-by-column. However, it need not be in the order of columns 1, 2, 3 etc. It can be any random order such as 2, 3, 1, etc.

3. The message thus obtained is the cipher text message of round 1.

4. Repeat steps 1 to 3 as many times as desired.

VERNAM CIPHER/ ONE TIME PAD

1. Treat each plain text alphabet as a number in an increasing sequence i.e. A = 0, B = 1, … Z = 25.

2. Do the same for each character of the input cipher text.

3. *Add* each number corresponding to the plain text alphabet to the corresponding input cipher text alphabet number.

4. If the sum thus produced is greater than 26, subtract 26 from it.

5. Translate each number of the sum back to the corresponding alphabet. This gives the output cipher text.

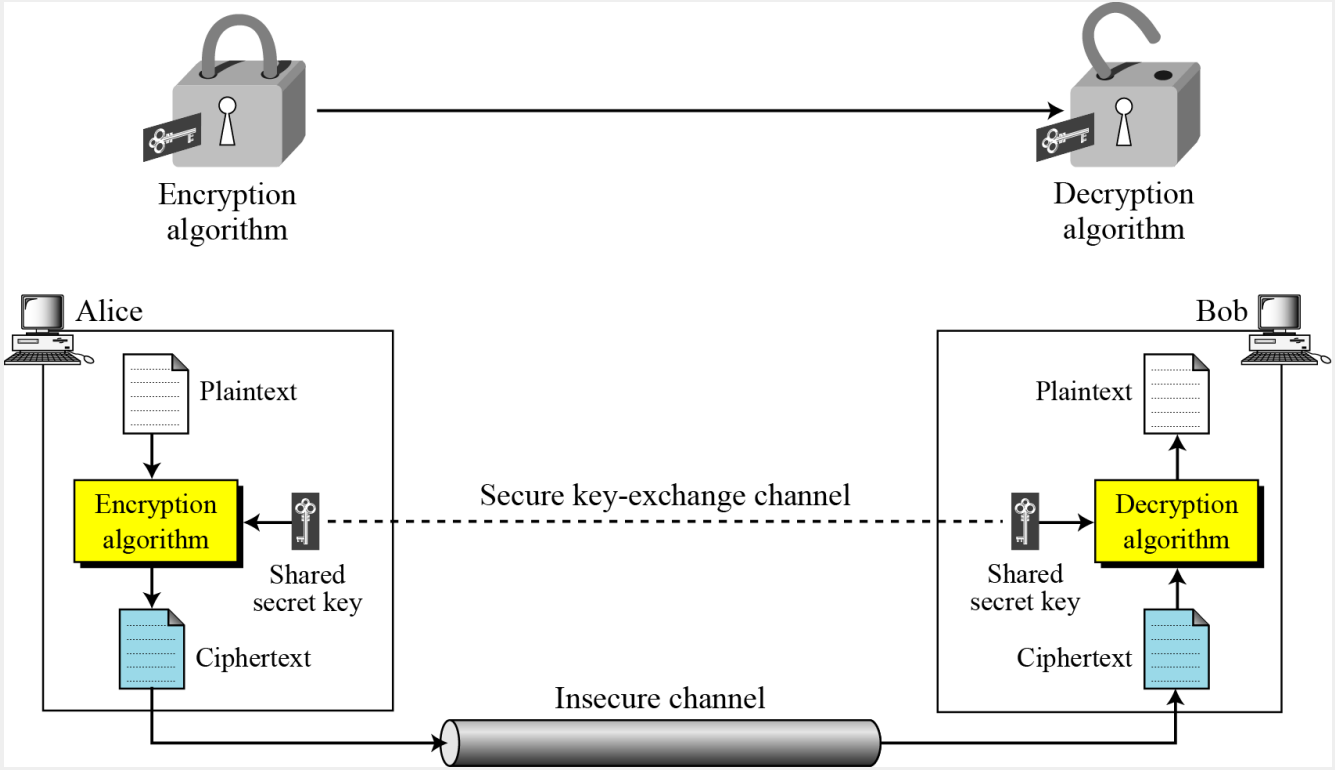
VERNAM CIPHER/ ONE TIME PAD

	H	E	L	L	O	message
	7	4	11	11	14	message
	X	M	C	K	L	Key/One time Pad
	23	12	2	10	11	Key
=	30	16	13	21	25	message + key
=	4	16	13	21	25	message + key (mod 26)
	E	Q	N	V	Z	ciphertext

ENCRYPTION ALGORITHMS: SYMMETRIC

- There are two general types of key-based algorithms: symmetric and asymmetric
- **Symmetric algorithms,** sometimes called conventional algorithms, are algorithms where the encryption key can be calculated from the decryption key and vice versa.
- In most symmetric algorithms, the encryption key and the decryption key are the same. These algorithms, also called secret-key algorithms.

ENCRYPTION ALGORITHMS: SYMMETRIC



ENCRYPTION ALGORITHMS: SYMMETRIC

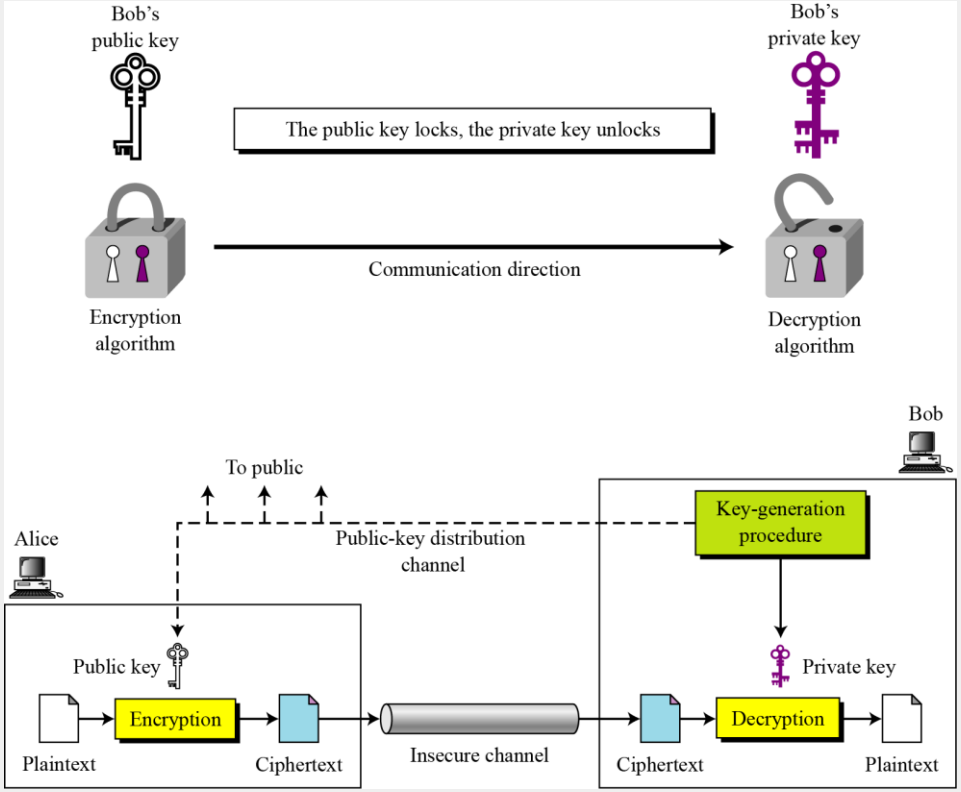
Disadvantages:

- Requires prior communication of the secret key between the sender and the receiver.
- The secret key have to be changed periodically.
- The number of keys in the key management process may be large.

ENCRYPTION ALGORITHMS: ASYMMETRIC

- ***Public-key algorithms*** also called ***asymmetric algorithms***, are algorithms where the key used for encryption is different from the key used for decryption.
- the decryption key cannot be calculated from the encryption key.
- the encryption key is often called the ***public key*****, and**
- the decryption key is often called the ***private key*****.**
- If A wants to send a message securely to B, he will encrypt the message using B's public key.
- Since B is the only user who know the matching secret key , only B can decrypt the message.

ENCRYPTION ALGORITHMS: ASYMMETRIC



PUBLIC-KEY APPLICATIONS

It can classify uses into 3 categories:

- ***Encryption/decryption*** (provide secrecy)
- ***Digital signatures*** (provide authentication)
- ***Key exchange*** (of session keys)

SYMMETRIC VS. ASYMMETRIC KEY CRYPTOGRAPHY

Characteristic	*Symmetric Key Cryptography*	*Asymmetric Key Cryptography*
Key Used	Same key for encryption & decryption	One key for encryption & different key for decryption
Speed	Very fast	Slower
Key agreement/exchange	A big problem	No problem at all
Size of encrypted text	Usually same or less than the original PT size	More than the PT size
No. of keys required as compared to participants	Equals about square of the no. of participants	Same as the no. of participants
Usage	Provide confidentiality. Cant be used for digital signature	Can also provide authentication and nonrepudiation. Used for digital signature
Example	DES, AES, Blowfish, RC4, RC5	RSA, Diffie-Hellman, El Gamal, ECC

DATA ENCRYPTION STANDARD (DES)

Data Encryption Standard

RSA

1. Choose two large prime numbers P and Q.

2. Calculate N = P x Q.

3. Select the public key (i.e. the encryption key) E such that it is not a factor of (P – 1) and (Q – 1).

4. Select the private key (i.e. the decryption key) D such that the following equation is true: (D x E) mod (P – 1) x (Q – 1) = 1

5. For encryption, calculate the cipher text CT from the plain text PT as follows: $CT = PT^E \bmod N$

6. Send CT as the cipher text to the receiver.

7. For decryption, calculate the plain text PT from the cipher text CT as follows: $PT = CT^D \bmod N$

RSA EXAMPLE

- Select primes p=11, q=3.
- n = pq = 11 * 3 = 33
- (p-1)(q-1) = 10 *2 = 20

Choose E=3

Check gcd (E, (p-1)(q-1)) = gcd(3, 20) = 1

(i.e. 3 and 20 have no common factors except 1)

- Compute D such that (D *E) mod (p-1)(q-1)=1
  i.e. compute D = e^{-1} mod (p-1)(q-1)= 3^{-1} mod 20
  i.e. find D such that 20 divides 3D-1.
  Simple testing (d = 1, 2, ...) gives d = 7
  Check: 3*7 - 1 = 20, which is divisible by (p-1)(q-1)
- Public key = (E,n) = (3, 33) Private key = (D,n) = (7,33).

RSA EXAMPLE

- Now say we want to encrypt the message m = 7,
  $CT = PT^E \bmod n$

  $= 7^3 \bmod 33$

  $= 343 \bmod 33 = 13.$

  Hence the cipher text c = 13.
- To decrypt we compute

  $PT = CT^D \bmod n$

  $= 13^7 \bmod 33 = 7.$

  $m = 13^7 \bmod 33 = 13^{(3+3+1)} \bmod 33 = 13^3.13^3.13 \bmod 33$

  $= (13^3 \bmod 33).(13^3 \bmod 33).(13 \bmod 33) \bmod 33$

  $= (2197 \bmod 33).(2197 \bmod 33).(13 \bmod 33) \bmod 33$

  $= 19.19.13 \bmod 33 = 4693 \bmod 33$

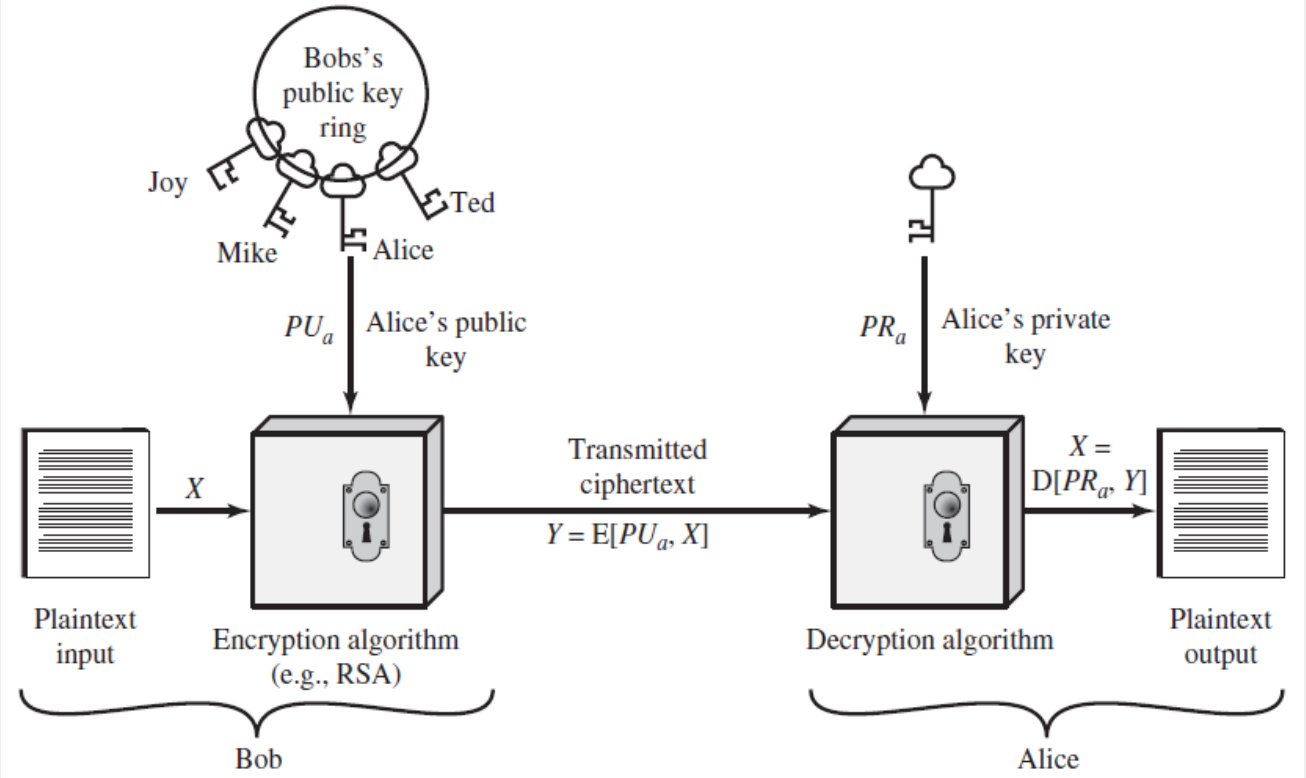
  $= 7$

DES VS. RSA

Method	DES	RSA
Approach	Symmetric	Asymmetric
Encryption	Faster	Slow
Decryption	Faster	Slow
Key distribution	Difficult	Easy
Complexity	O(Log N)	O (N3)
Security	Moderate	Highest
Nature	Closed	Open
Inherent Vulnerabilities	Brute Forced, Linear and differential cryptanalysis attack	Brute Forced and Oracle attack
Vulnerabilities cause	Weak key usage	Weak implementation
Secure Services	Confidentially	Confidentially, integrity, non repudiation

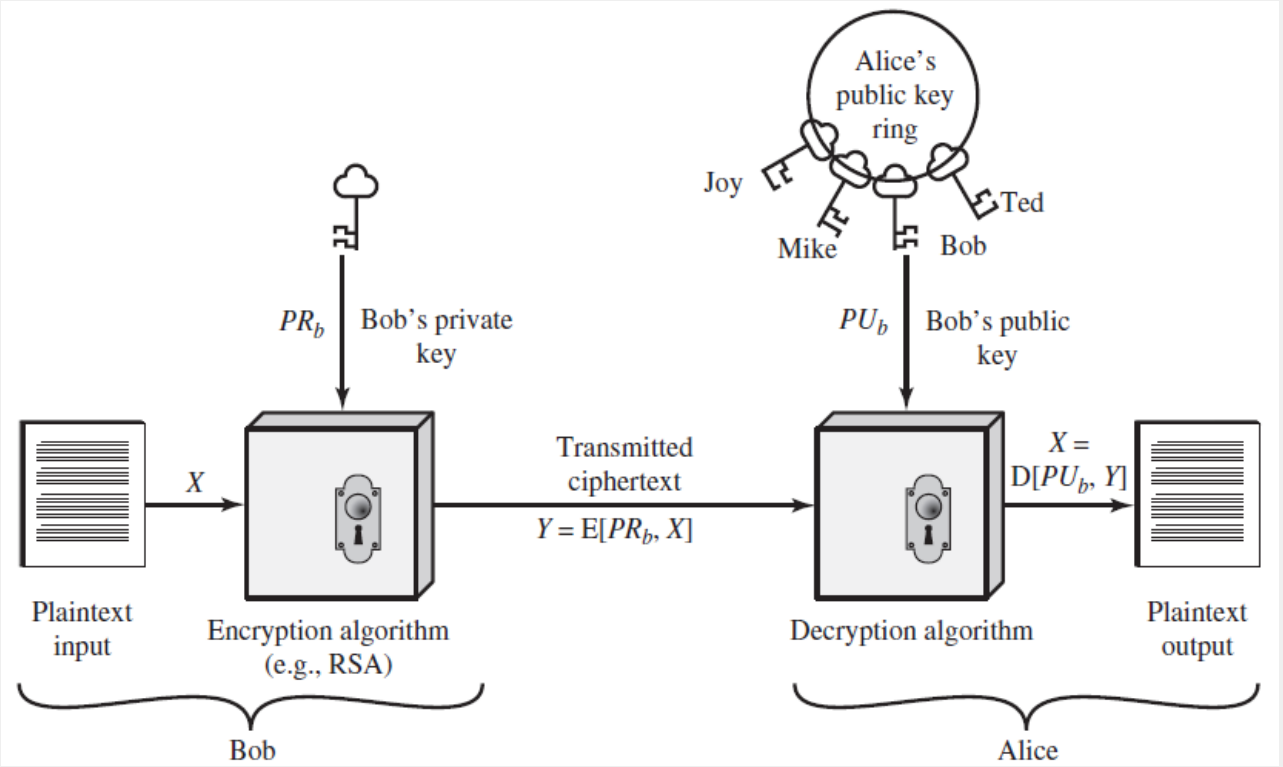
PUBLIC KEY ENCRYPTION

Encryption with Public key



PUBLIC KEY ENCRYPTION

Encryption with Private key



STREAM CIPHER

BLOCK CIPHER

KEY MANAGEMENT

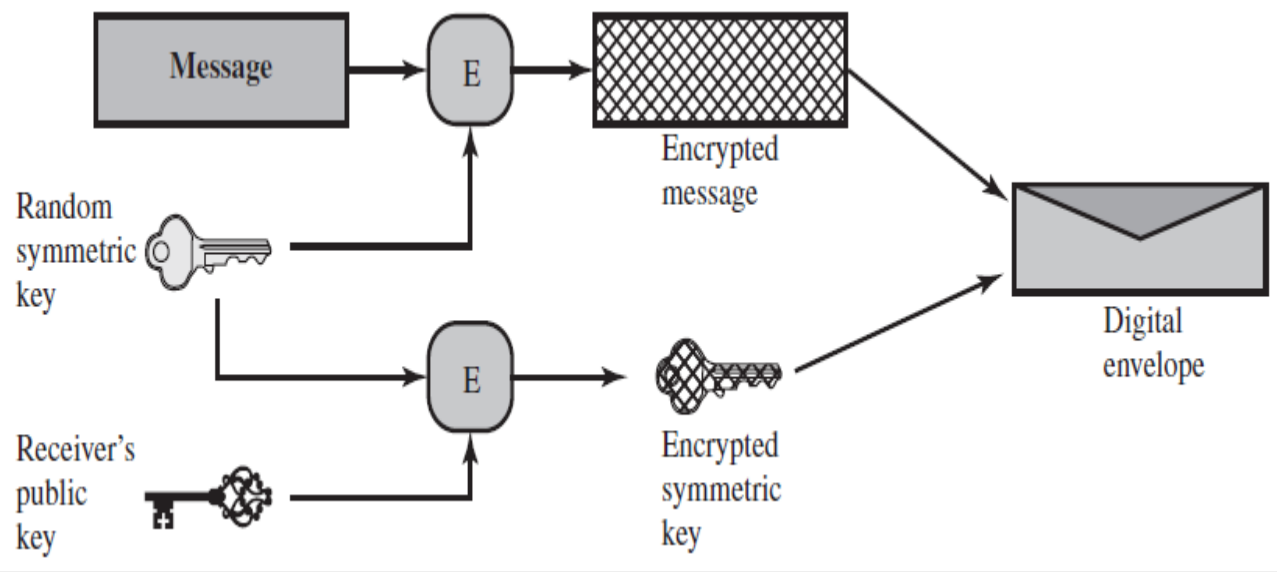
Key management refers to the

- Distribution of cryptographic keys
- Mechanisms used to bind an identity to a key
- The generation, maintenance, and revoking of such keys.

SYMMETRIC-KEY EXCHANGE

Digital envelopes

- Prepare a message.
- Generate a random symmetric key that will be used this one time only

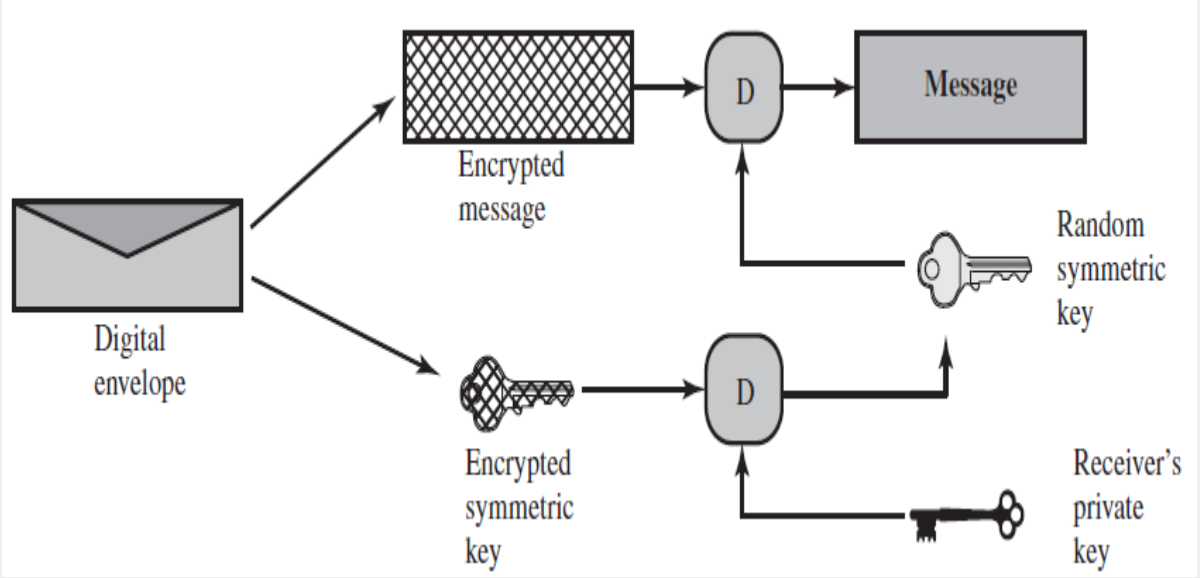


Creation of a Digital Envelope

SYMMETRIC-KEY EXCHANGE

Digital envelopes

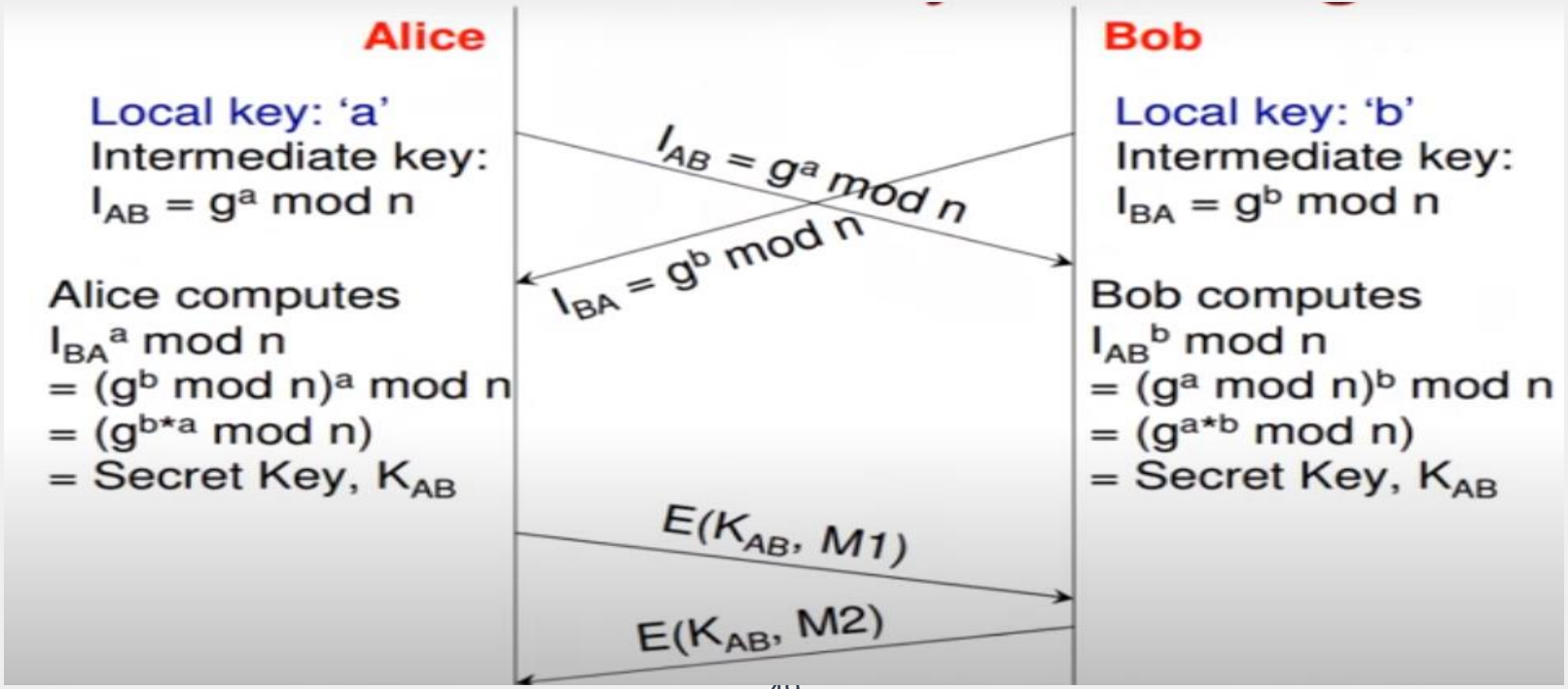
- Encrypt that message using symmetric encryption the one-time key.
- Encrypt the one-time key using public-key encryption with Alice's public key.
- Attach the encrypted one-time key to the encrypted message and send it to Alice.



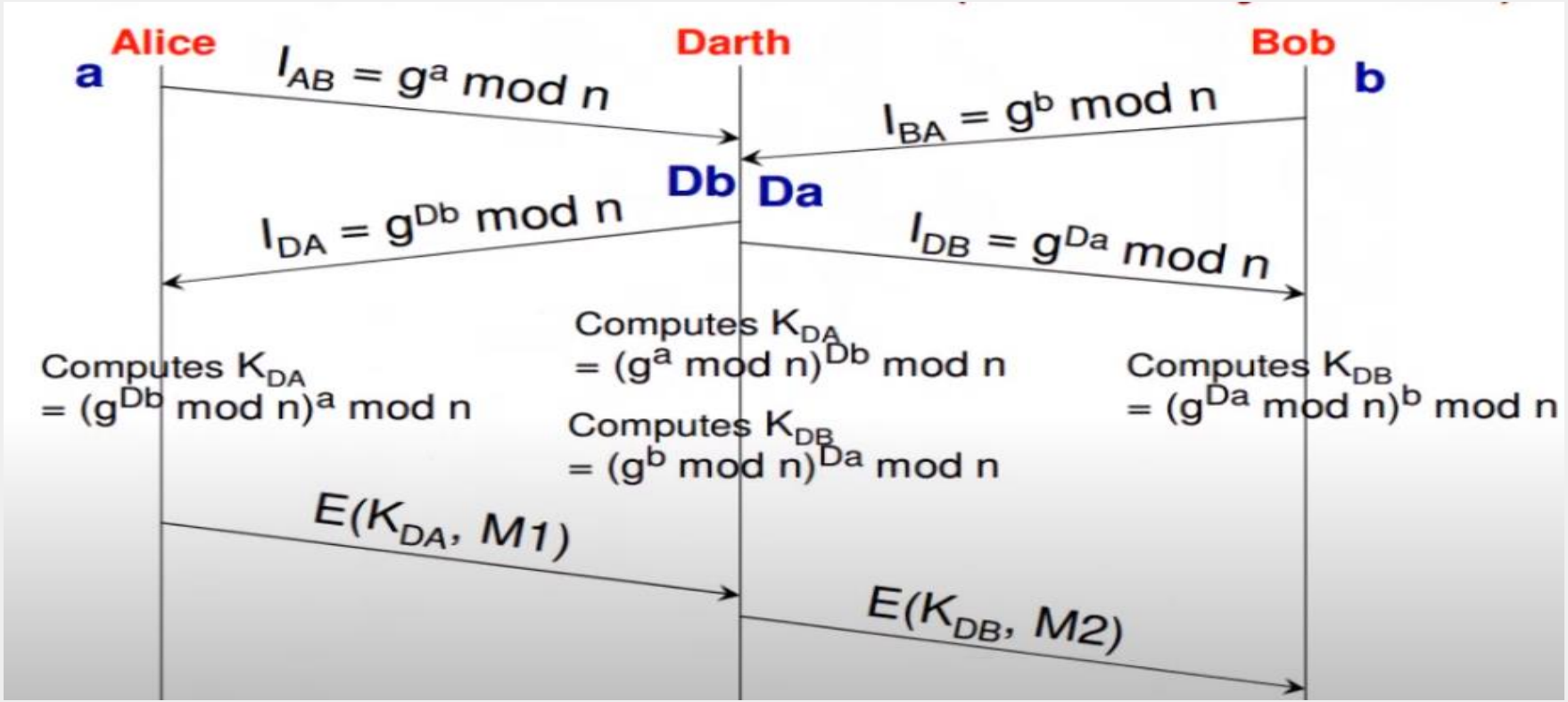
Opening a Digital Envelope

SYMMETRIC-KEY EXCHANGE

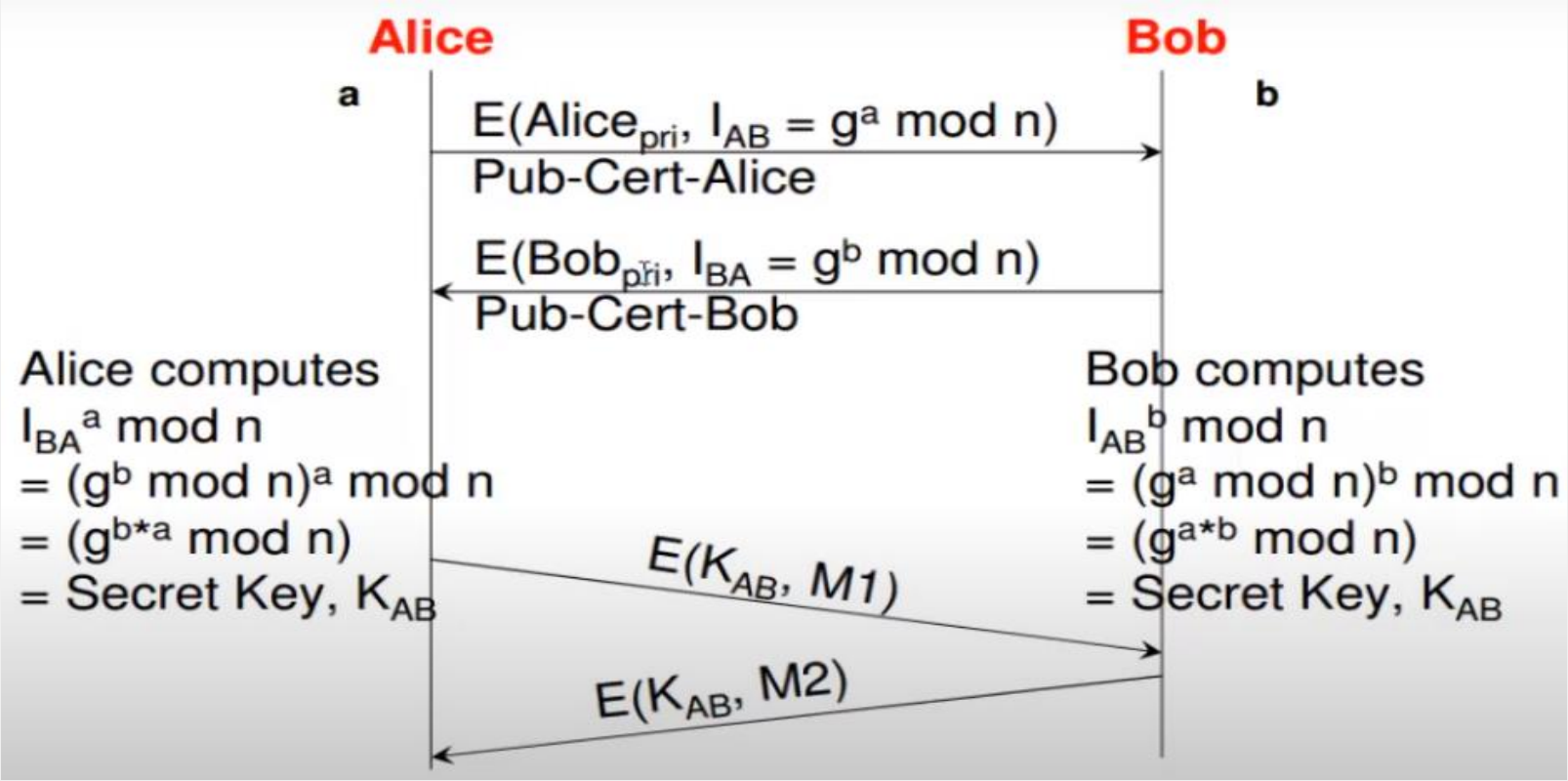
Diffie-Hellman Key Exchange



MAN-IN-MIDDLE ATTACK DIFFIE-HELLMAN KEY EXCHANGE



STATION TO STATION PROTOCOL



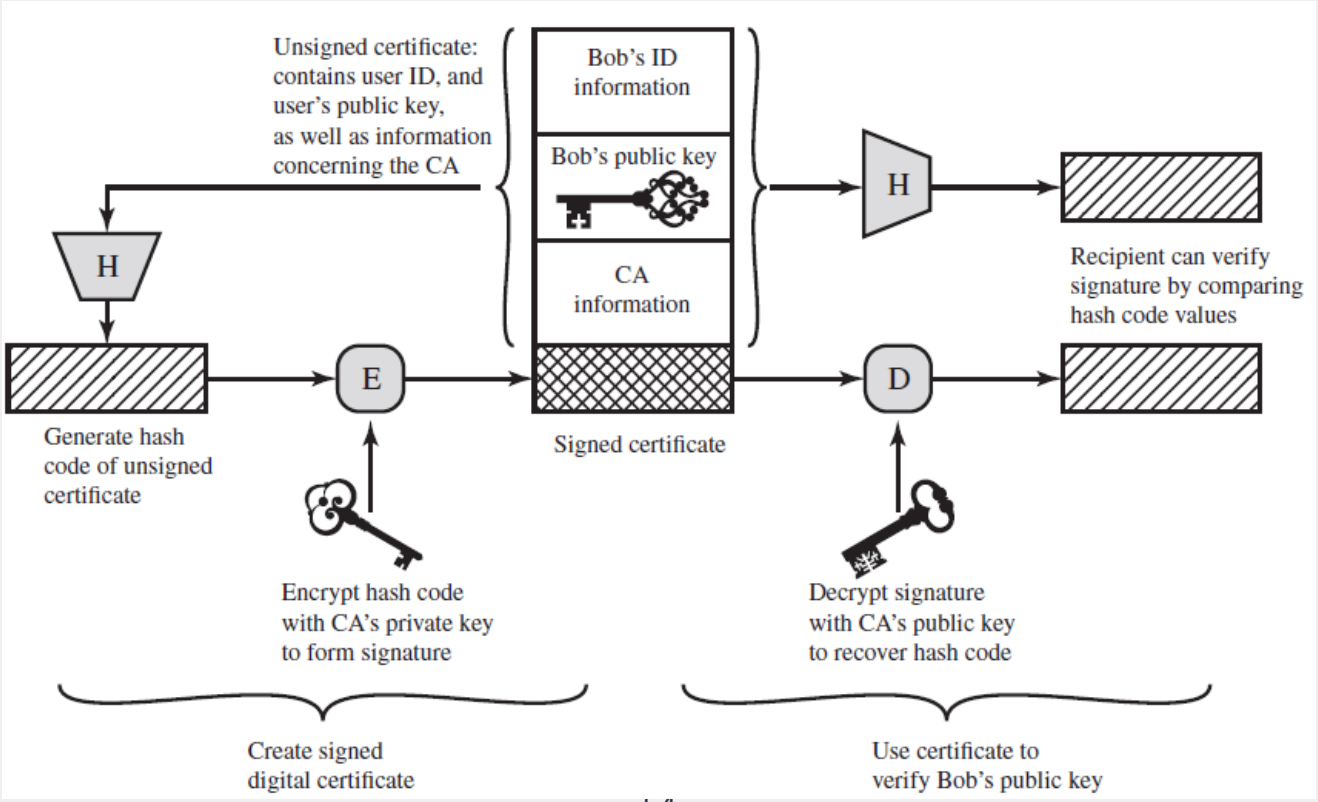
PUBLIC-KEY CERTIFICATE USE

1. User software (client) creates a pair of keys: one public and one private.
2. Client prepares an unsigned certificate that includes the user ID and user's public key.
3. User provides the unsigned certificate to a CA in some secure manner. This might require a face-to-face meeting, the use of registered e-mail, or happen via a web form with e-mail verification.
4. CA creates a signature as follows:
   - CA uses a hash function to calculate the hash code of the unsigned certificate. A hash function is one that maps a variable-length data block or message into a fixed-length value called a hash code, such as SHA family.
   - CA encrypts the hash code with the CA's private key.

PUBLIC-KEY CERTIFICATE USE

5. CA attaches the signature to the unsigned certificate to create a signed certificate.
6. CA returns the signed certificate to client.
7. Client may provide the signed certificate to any other user.
8. Any user may verify that the certificate is valid as follows:
   - User calculates the hash code of certificate (not including signature).
   - User decrypts the signature using CA's known public key.
   - User compares the results of (a) and (b). If there is a match, the certificate is valid.

PUBLIC-KEY CERTIFICATE USE



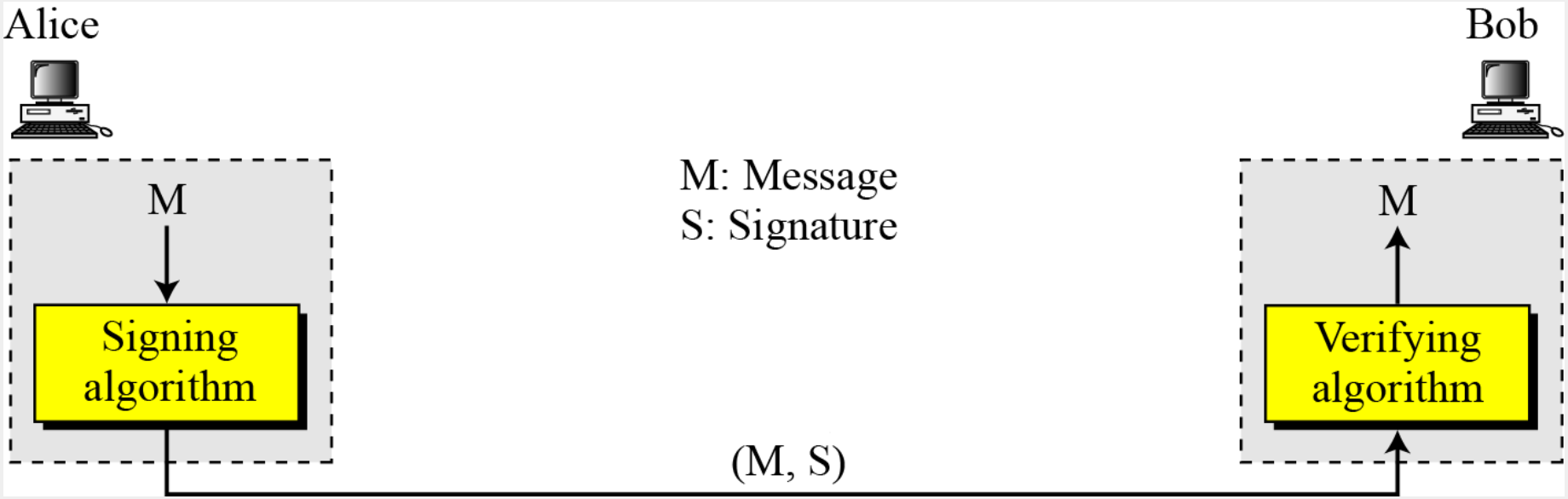
DIGITAL SIGNATURE

- The sender uses a signing algorithm to sign the message. The message and the signature are sent to the receiver. The receiver receives the message and the signature and applies the verifying algorithm to the combination. If the result is true, the message is accepted; otherwise, it is rejected.

- *"A digital signature is a construct that authenticates both the origin and contents of a message in a manner that is provable to a disinterested third party."*

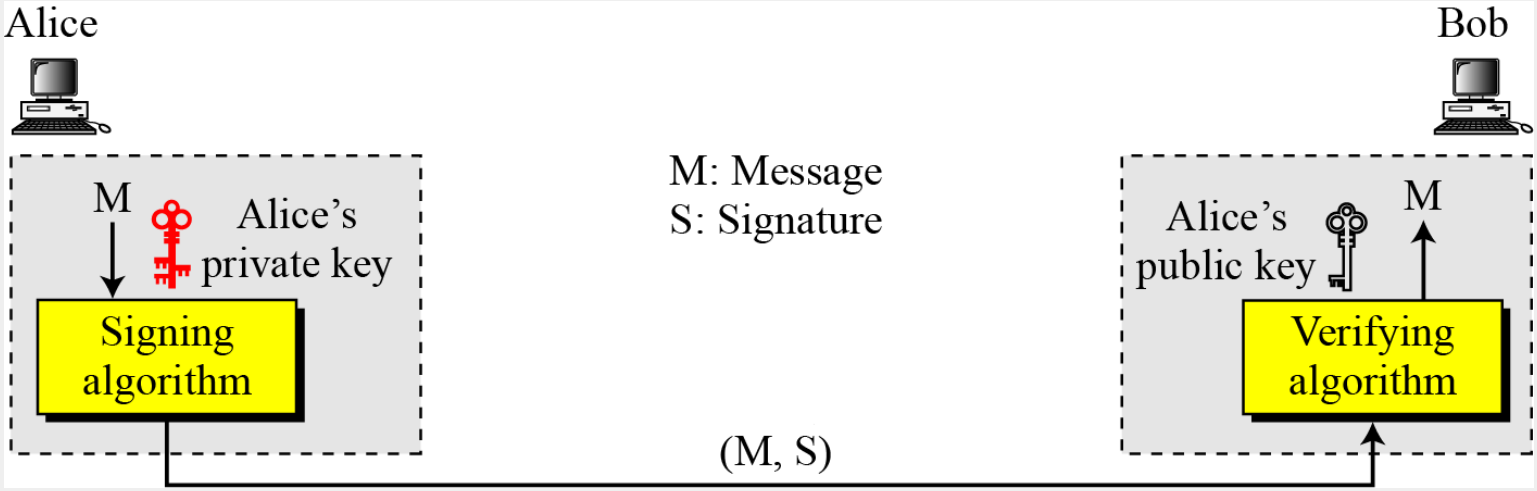
DIGITAL SIGNATURE

Digital Signature Process



DIGITAL SIGNATURE

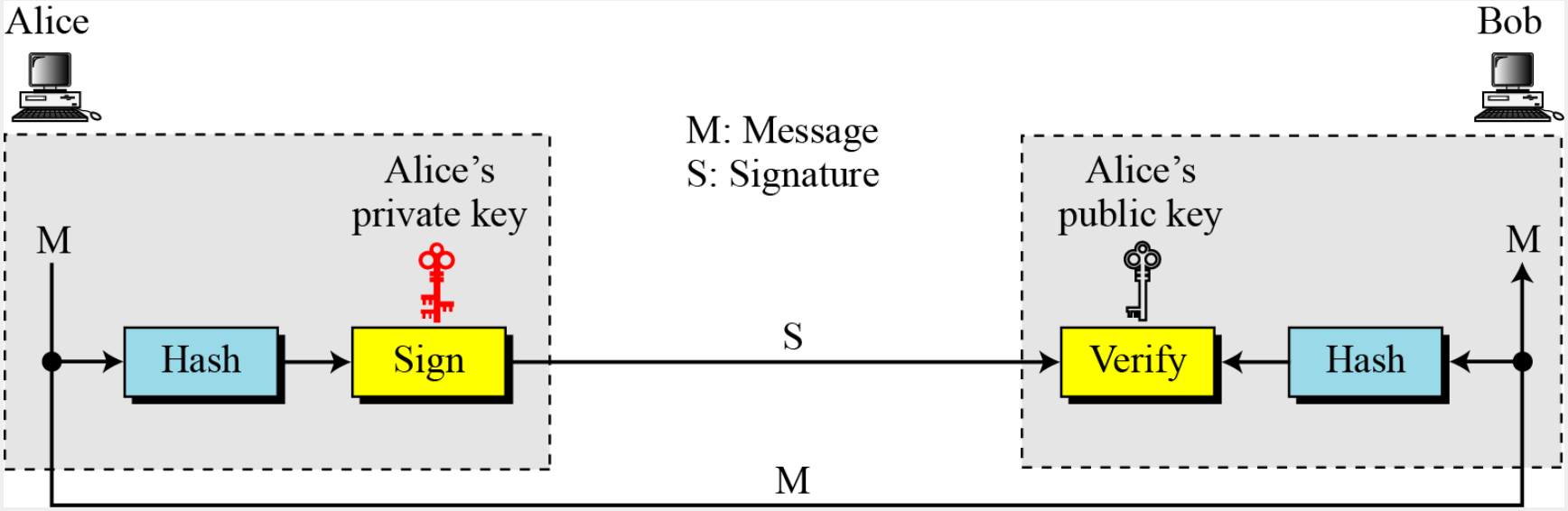
Adding key to the digital signature process



A digital signature needs a public-key system.
The signer signs with her private key; the verifier verifies with the signer's public key.

DIGITAL SIGNATURE

Signing the digest



DIGITAL SIGNATURE

Message Authentication

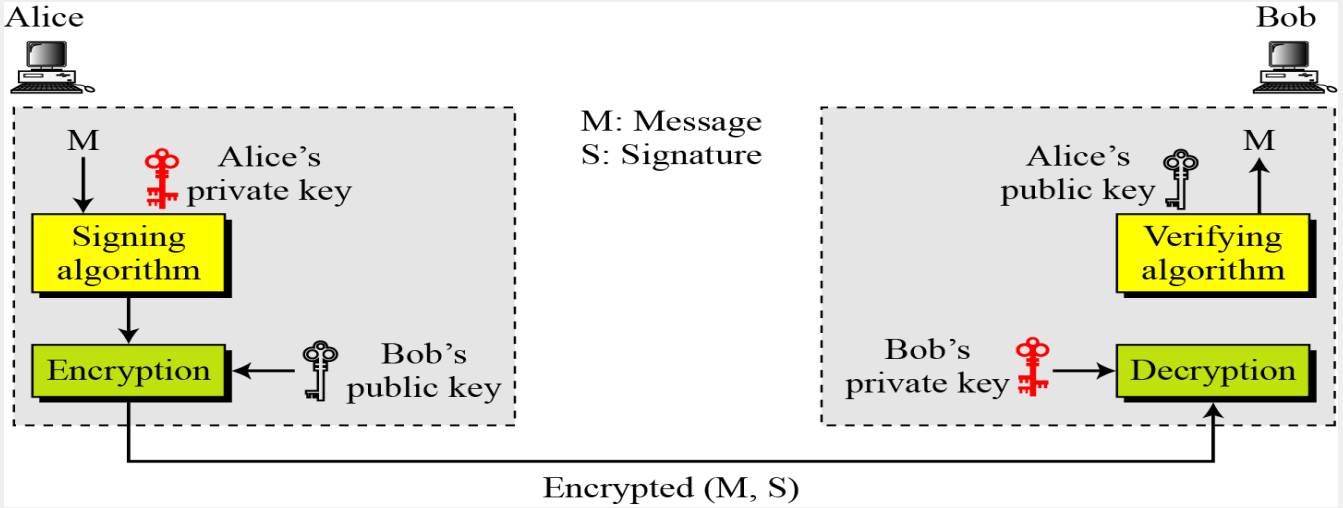
- A secure digital signature scheme, like a secure conventional signature can provide message authentication.

Message Integrity

- The integrity of the message is preserved even if we sign the whole message because we cannot get the same signature if the message is changed.

DIGITAL SIGNATURE

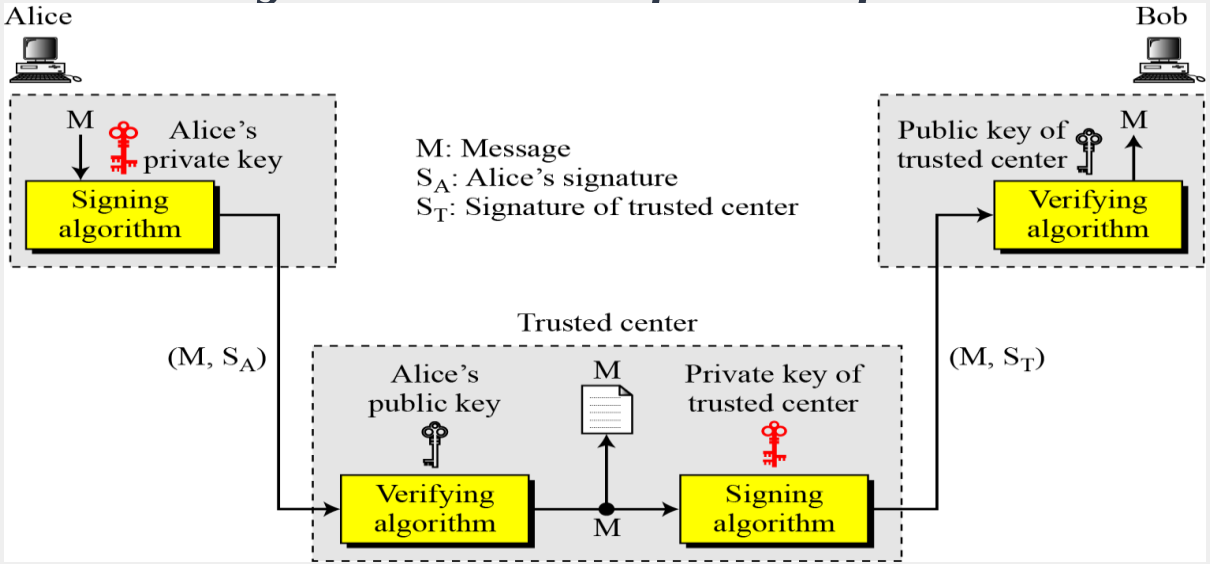
Adding confidentiality to a digital signature scheme



A digital signature does not provide privacy.
If there is a need for privacy, another layer of encryption/decryption must be applied.

DIGITAL SIGNATURE

Using a trusted center for nonrepudiation



Nonrepudiation can be provided using a trusted party.

KEY MANAGEMENT

Notation: $X \rightarrow Y : \{ Z \} k$

indicates entity X sends entity Y a message Z enciphered with key k.

- Subscripts to keys indicate to whom the keys belong, and are written where multiple keys are in use.
- k_{Alice} and k_{Bob} refer to keys belonging to Alice and Bob,
- If Alice and Bob share a key, that key will be written as $k_{Alice, Bob}$ when the sharers are not immediately clear from the context.
- In general, k represents a secret key (for a classical cryptosystem), e a public key, and d a private key (for a public key cryptosystem).
- If multiple messages are listed sequentially, they are concatenated and sent. The operator a || b means that the bit sequences a and b are concatenated.

SESSION KEYS AND INTERCHANGE KEYS

- An *interchange key* is a cryptographic key associated with a principal to a communication.

  It is used for all sessions. It changes independently of session initiation and termination.

- A *session key* is a cryptographic key associated with the communication itself.

  e.g. Alice has a cryptographic key used specifically to exchange information with Bob. This key does not change over interactions with Bob. if Alice communicates twice with Bob she does not want to use the same key to encipher the messages. It is discarded when the session ends. Hence, the name "**session key**."

KEY EXCHANGE

- The goal of key exchange is to enable Alice to communicate secretly to Bob, and vice versa, using a shared cryptographic key.
- Solutions to this problem must meet the following criteria.
  - The key that Alice and Bob are to share cannot be transmitted in the clear. Either it must be enciphered when sent, or Alice and Bob must derive it without an exchange of data from which the key can be derived.
  - Alice and Bob may decide to trust a third party.
  - The cryptosystems and protocols are publicly known. The only secret data is to be the cryptographic keys involved.

CLASSICAL CRYPTOGRAPHIC KEY EXCHANGE AND AUTHENTICATION

- Suppose Alice and Bob wish to communicate.
- If they share a common key, they can use a classical cryptosystem. ***But how do they agree on a common key?***
- If Alice sends one to Bob, Eve the eavesdropper will see it and be able to read the traffic between them.

Any suggestions??

- To avoid this bootstrapping problem, classical protocols rely on a trusted third party, Cathy.
- Alice and Cathy share a secret key, and Bob and Cathy share a (different) secret key.
- The goal is to provide a secret key that Alice and Bob share.

SESSION KEYS AND INTERCHANGE KEYS

- Alice →Cathy: { request for session key to Bob }k_{Alice}
- Cathy→ Alice: { $k_{session}$ }k_{Alice} || { $k_{session}$ }k_{Bob}
- Alice →Bob: { $k_{session}$ }k_{Bob}
- Bob now deciphers the message and uses $k_{session}$ to communicate with Alice.
- Any Drawbacks??

SESSION KEYS AND INTERCHANGE KEYS

- However, Bob does not know to whom he is talking.
- Assume that Alice sends Bob a message

  (such as "Deposit $500 in Swiss bank account today") enciphered under $k_{session}$.
- If Eve records the second message in the exchange above, and the message enciphered under $k_{session}$, she can send Bob the message $\{ k_{session} \}k_{Bob}$ followed by the message enciphered under $k_{session}$. Bob will not know who is sending it.
- Avoiding problems such as this replay attack adds considerable complexity.
- One of the best-known protocol that provides little defense against replay attack is ***the Needham-Schroeder protocol*** .

NEEDHAM-SCHROEDER PROTOCOL

- Alice →Cathy : { Alice || Bob || $rand_1$ }

  Alice sends a message to Cathy identifying herself and Bob, telling Cathy she wants to communicate with Bob.

- Cathy→ Alice : { Alice || Bob || $rand_1$ || $k_{session}$ || {Alice || $k_{session}$} k_{Bob} } k_{Alice}

  Cathy generates $k_{session}$ and sends back to Alice a copy encrypted under k_{Bob} for Alice to forward to Bob and also a copy for Alice. The nonce assures Alice that the message is fresh and that Cathy is replying to that particular message and the inclusion of Bob's name tells Alice who she is to share this key with.

- Alice →Bob : { Alice || $k_{session}$ } k_{Bob}

  Alice forwards the key to Bob who can decrypt it with the key he shares with Cathy, thus authenticating the data.

- Bob →Alice : { $rand_2$ } $k_{session}$

  Bob sends Alice a nonce encrypted under $k_{session}$ to show that he has the key

- Alice →Bob : { $rand_2$ -1 }$k_{session}$

  Alice performs a simple operation on the nonce, re-encrypts it and sends it back verifying that she is still alive and that she holds the key

STUDY MATERIALS

- Matt Bishop,“ Introduction to Computer Security ”, Pearson Education,2005.
- Atul Kahate, “Cryptography and Network Security”, third edition

Please Note : PPT’s are for Reference Only , NOT AS A STUDY MATERIAL

Contact

Krishna Samdani

krishna.samdani@nmims.edu

Mobile- 9920407045

Seating – 1A Faculty Area MPSTME Building

Thanks!

Any questions?